

slide : 6

structuring system Process

Requirements

Process modeling

↳ graphically represent processes that: Capture, store and distribute data between a system and its environment and among system components.

Data Flow diagram (DFD)

↳ graphically illustrate movement of data between external entities and process and data in system.

Process modeling to system

- 1) utilize info. gathered during determining requirements.
- 2) structure of data modeled + processes.

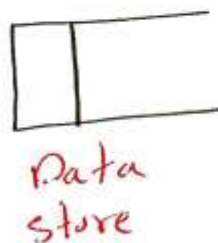
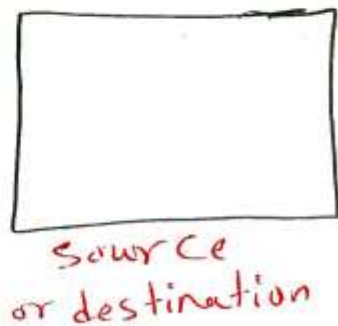
DFD of new & current logical system

a) technology independent

b) show data flows, structure & functional requirements of new sys.

→ DFD of current physical system: Enables analysts to understand current system

DFD Symbols



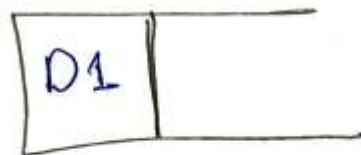
1) Data Flow (→)

- a) describe data that are in motion & moving as a unit from place to another.
- b) select meaningful name to represent data.

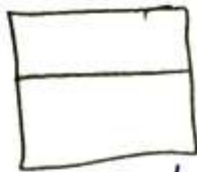
2) Data store (□)

- a) describe data at rest.
- b) may represent data in:
 - 1. Physical file folder or notebook.
 - 2. Computer-based file (database, figures...)

- c) each data store take a code name not repeated.



3) Process



- a) describe work or action or task performed on data that they transformed.
- b) Choose suitable name to it "verb" (like: update data)
- c) each process has a number.

- ### 4) Source/sink
- ()
- a) describe origin and/or destination of data.
 - b) some times referred to external entity.
 - c) we get external data from source.
 - d) its names is noun (Bank, Customer, ...)

DFD definitions

*Context Diagram

- ~~data~~ DFD of the scope of an organizational system. that shows system (boundaries, external entities) that interact with system.
- no sources data store on context diagram.

* Level-0 - diagram

→ DFD that represents system's (main processes, data flows, and data stores) at high level of detail.

DFD rules

- 1) Input are different ~~that~~ than outputs.
- 2) objects have a unique name.

3) Process

- a. no process can have only outputs.
- b. no " " " " " " inputs.
- c. a process has a verb phrase label.

4) Data store.

- a. data cannot be moved direct from store to another
- b. " " " " " " from outside source to data source.
- c. " " " " " " from data store to data sink.
- d. data store has noun phrase label.

5) Source / sink:

- a. data cannot move directly from source to sink.
- b. source / sink has a noun phrase label.

6) Data Flow

- a. It has only one direction of flow.
 - b. data flow cannot go directly back to process it leaves.
 - c. data flow to data store means update.
 - d. " " From " " " retrieve or use.
 - e. " " has noun phrase label.
-

*Functional decomposition

- a. begin with high level diagram and then decompose each high process to simple ones.
- b. Functional decomposition: is the act of going from one single system to many component processes.
- c. lowest level called primitive DFD.

*Level - N Diagram

- DFD that is the result of n-nested decompositions of a series of subprocesses on level-0 diagram.

Balancing DFDs

(يَحْفَظ)

a. When decomposing DFD, you must ~~Conserve~~ ^{Conserve} inputs to outputs from process at next level of decomposition.

→ process must have same I/P & O/P when it is decomposed (مقسّم) to next level.

* This conservation is called balancing.

Types of DFDs

1) Current Physical

a. Process label include identification of the technology used to process the data.

b. data flow & data stores are labeled with actual name of physical media which data flow is stored.

2) Current Logical

a. Physical aspects of system are removed as much as possible.

b. Current system is reduced to data and processes that transform them.

3) New logical

- a. includes additional functions.
- b. inefficient data flows are reorganized.
- c. neglected functions are removed.

4) New Physical

- a. represents physical implementation of new system.
- b. analyst make decision on which parts of system will be automated and which will be kept manual.

Guidelines for drawing DFD

* Completeness

- a. It must include all components necessary to system.
- b. Each component is fully described.

* Consistency

- means information contained on one level of a set of nested DFDs is also included on other levels

* Timing

- a. time is not represented well on DFDs.
- b. Best to draw DFDs as if system has never started and will never stop.

* Iterative development

→ Analyst should expect to redraw diagram several times till reaching the end.

* Primitive DFDs

- a. lowest logical level of decomposition.
- b. Decision has to be made when to stop decomposition.

Rules for stopping decomposition

- a. When every process has been reduced to single decision, database operation.
- b. When each data store represents data about single entity.
- c. When system user does not care to see any more detail.